August 1988

Report No. ONR-111

(4)

ONR Final Report: Implementation and Evaluation of Automated Maintainability Assessment

Douglas M. Towne
Mark C. Johnson
Nicholas A. Bond, Jr.

BEHAVIORAL TECHNOLOGY LABORATORIES
Department of Psychology
University of Southern California

Sponsored by

Perceptual Science Division
Office of Naval Research

Under Contract No. N00014-86- K-0793
Project Code 4429004

ONR Final Report:

Implementation and Evaluation of Automated Maintainability Assessment

Douglas M. Towne
Mark C. Johnson
Nicholas A. Bond, Jr.

August 1988

Technical Report No. ONR-111

Behavioral Technology Laboratories
University of Southern California

Sponsored by

Perceptual Science Division
Office of Naval Research

Under Contract No. N00014-86- K-0793
Project Code 4429004

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3 DISTRIBUTION / AVAILABILITY OF REPORT "Approved for Public Release: Distribution Unlimited"
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Technical Report No. ONR-111	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION University of Southern Calif. Behavioral Technology Laboratories 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Perceptual Science Division, Code 1142 PS
6c. ADDRESS (City, State, and ZIP Code) 1845 South Elena Ave., 4th Flr. Redondo Beach, CA 90277	7b. ADDRESS (City, State, and ZIP Code) 800 North Quincy St. Arlington, VA 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0793
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62757N; PROJECT NO. RS 34H20; TASK NO 15; WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Unclassified ONR Final Report: Implementation and Evaluation of Automated Maintainability Assessment

12. PERSONAL AUTHOR(S)
Douglas M. Towne, Mark C. Johnson, Nicholas A. Bond Jr.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 8/15/86 TO 8/14/88	88/08/15	35

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	PROFILE, Predicting Maintainability, Fault Isolation, Intelligent Maintenance Training System, Projecting Maintenance Workload

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is the final report for research performed under contract N00014-86-K-0793. ~~The~~ research is part of a multi-disciplinary computer program concerned with design for maintainability. The primary objectives of this component have been 1) to integrate the tools for analyzing maintainability developed earlier into a representative and commonly-used CAD/CAE (computer-aided design/computer-aided engineering) system, and 2) to evaluate the performance of the analysis technique, both from the standpoint of technical accuracy and ease of use.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Gerald S. Malecki	22b. TELEPHONE (Include Area Code) (202) 696-4289 — 22c. OFFICE SYMBOL ONR 1142PS

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Abstract

This is the final report for research performed under contract N00014-86-K-0793. The research is part of a multi-disciplinary program concerned with design for maintainability. The primary objectives of this component have been 1) to integrate the tools for analyzing maintainability developed earlier into a representative and commonly-used CAD/CAE (computer-aided design,computer-aided engineering) system, and 2) to evaluate the performance of the analysis technique, both from the standpoint of technical accuracy and ease of use

Contents

Figures

Tables

Acknowledgements

This research was sponsored by the Perceptual Science Program, Office of Naval Research, Gerald S. Malecki serving as scientific officer. We wish to sincerely thank them for their support of this work. We also thank Martin Tolcott, formerly of the Office of Naval Research, for his support and encouragement during the conceptual formation phases of this research.

The installation and evaluation of Profile at NOSC was made possible by the active cooperation of Mr. Mel Nunn, Head of Computer Integrated Engineering, Naval Ocean Systems Center, San Diego, and by Mr. Mike Dwyer of that same office who provided and arranged access to technical expertise and served as the central design user of Profile.

We also thank the engineers and designers at NOSC who supplied design specifications for the evaluations, namely Mr. Alex Amperan, Mr. Perry Partow, Mr. Mike Juniper, Mr. Lee Skeen, and Mr. Paul Wong. We thank Dr. Nick Kanopoulos of Research Triangle Institute for his contribution of the Doppler Filter circuit and associated documentation.

We sincerely thank the people of Mentor Graphics Corporation for contributing their CAD system, IDEA, for this work.

1 Introduction

This is the final report for research performed under contract N00014-86-K-0793. The research is part of a multi-disciplinary program concerned with design for maintainability. The primary objectives of this component have been 1) to integrate the tools for analyzing maintainability developed earlier into a representative and commonly-used CAD/CAE (computer-aided design,computer-aided engineering) system, and 2) to evaluate the performance of the analysis technique, both from the standpoint of technical accuracy and ease of use by system designers.

1.1 Background

A goal of this country's developers of military systems for the final two decades of this century is to shape their management and technical forces according to the discipline called Integrated Diagnostics (ID). The ID discipline calls for communication and planning functions about diagnostic requirements that are performed earlier in the design process and more frequently than ever before. When ID precepts are followed perfectly, all diagnostic requirements of a system are anticipated and addressed long before the system takes physical shape. While this objective may be difficult to achieve perfectly, it represents a goal that must be approached if future generations of complex systems are to be maintained effectively.

Adherence to the ID discipline will depend critically upon the ability of designers to anticipate diagnostic requirements and to effectively and quantitatively measure the success with which alternative design concepts address those requirements. As we view it today, such capabilities are not in the hands of the design community for performing these crucial functions.

A promising technique for analyzing the maintainability characteristics of a system under design was developed during the early- to mid-1980's under funding from the Office of Naval Research (Towne and Johnson, 1987). This technique, termed Profile, simulates the diagnosis of sample failures in a specified system, and it generates a testing sequence for each sample fault that is representative of the testing that would be performed by a qualified technician. The diagnostic strategy employed is a generalized approach aimed at minimizing a combined function of repair time and spares consumption.

When applied to a specific design specification, the Profile diagnostic strategy is sensitive to the internal architecture of the circuitry, as it affects the propagation and observability of abnormal effects; to the physical design, including the packaging and modularization of subsystems; and to the design of the diagnostic interface, including the front panel, the complement of documented test points, the extent and capabilities of automated testing functions, and peripheral testing and tooling provisions. For each failure analyzed, Profile generates a detailed action sequence of testing, adjusting, disassembly, replacement, and reassembly operations, in the order that a rational, well-trained technician might follow.

The time to perform the testing sequence for each sample fault is obtained by retrieving standard predetermined times for the pertinent diagnostic operations from a data bank. Typical predefined operations include such diagnostic activities as loosening a bolt, making an oscilloscope reading, observing a meter, and replacing a circuit board. When applied to a sample of failures, the Profile technique yields a distribution of predicted repair times, the mean of which is an estimate of Mean Time To Repair (MTTR). In addition to these quantitative measures, the technique yields projections of the kinds of diagnostic actions that will be performed, the allocation of diagnostic time and workload to the various functions and tests, measures of utility of various diagnostic design features, and projected rates of false replacements.

1.2 Objectives

Prior to the work described here, Profile had only been applied within a research and development environment. The devices submitted for Profile analysis were selected by the developers to test particular capabilities of the system. Furthermore, there was not a formal and automated interface between the CAD-derived representation of a system design and Profile.

As a result, a number of relatively complex operations were required to convert CAD-derived specifications into input forms that were both compatible with, and sufficient for, Profile analysis. The fidelity and value of the Profile projections, however, were sufficiently high to warrant development of the application process beyond that which would normally be pursued in a research and development environment.

The objectives of this work were, therefore, 1) to develop a formal software interface between a typical CAD system and Profile that would allow designers to invoke a Profile analysis of systems under design at any intermediate stage of development, 2) to develop well-defined procedures for applying Profile to very large systems, in which detailed CAD representations cannot reasonably be combined into a single data form, and 3) to install the technique at Naval Oceans System Center (NOSC) and determine the ease with which NOSC engineers and designers could apply the technique to systems of their choosing and the practical fidelity of the maintainability projections obtained.

2 Developments

The system developed under this research effort is a combination of commercial CAD/CAE modules and processes developed especially for maintainability analysis. The commercial CAD systems employed in the development environment were as follows:

1. Mentor Graphics' IDEA computer aided design system. This CAD system is among the more widely used software for computer aided design, and it was available both at NOSC where the evaluation was conducted and at the University of Southern California. The schematic capture program (NETED, for NETwork EDitor) within IDEA is a graphics editor that interacts with the designer to create schematic circuit diagrams. Transparently to the user, it also produces an underlying data structure, called a design file, that provides a computer-readable form of the design.

   Associated Mentor Graphics modules employed were: 1) SYMED, the symbol editor for creating new objects in the Mentor Graphics object library, and 2) EXPAND, a program that converts the design file from the NETED format to a 'flattened' form that is accessible to external programs (netlisters) that operate upon the design data.

2. a circuit simulation program, called ANDI (for ANalog and DIgital simulation), developed by Silvar-Lisco Corporation. ANDI simulates circuits that are mixtures of both analog and digital elements. This CAE tool was selected because it handles mixed circuits without requiring artificial specifications from the user to bridge the analog and digital portions.

2.1 Development of the CAD-Profile Interface

Programs were developed that allow the designer to call for a maintainability analysis of a design on the CAD workstation. The objective was to produce a complete system that would respond to a single command by the designer to automatically analyze the maintainability characteristics of the current design. These programs operate upon the data structures, called design files, that underlie the graphical representation of the design as it appears on the screen of the CAD workstation. Fortunately, the design files for commercial CAD systems are well documented and accessible to external programs.

Processes

The processes that yield maintainability assessments of a design are performed by a combination of commercially available CAD/CAE programs, the Profile system developed under previous ONR-sponsored projects, and special programs developed under this research contract.

The steps that are automatically executed in response to the "Profile" command are as follows:

1. for each object in the design, the functional description of the object is obtained from the CAD component library, and substituted into the raw design file.

2. the design file is converted from its original hierarchical form to a non-hierarchical (flattened) form.

3. the circuitry is simulated, producing the normal signal values at all nodes in the network.

4. a model of a failed component is substituted into the design file

5. the circuitry is simulated under the failed condition; the signal characteristics at each node in the network are recorded.

6. steps 4 and 5 are repeated for all failures of interest

7. for each failure simulated, the normality of each test result is determined by comparing the signal value under normal conditions to the value under the failed condition; a compacted file of normal and abnormal fault effects is written.

8. a model of diagnostic performance, Profile, is executed for each simulated failure, to determine the maintainability implications of the design.

2.2 Data Conversion Programs Developed

The conversion, simulation, and analysis steps are shown in Figure 1. The programs developed to accomplish the process are now described.

LOADLIB. A program that converts the sequential Mentor Graphics object library (containing data about the components' characteristics and internal connectivity) to a random-access file, so that a fault-insertion program can retrieve models of failed objects.

NETANDI. A program that converts the design file created with the Mentor Graphics schematic capture program into a form compatible with the ANDI circuit simulator.

FAILNET. A program that sequentially substitutes models of failed components into the netlist form produced with NETANDI. Failures simulated were of three types: 1) catastrophic failure of analog components, simulated by deleting the component

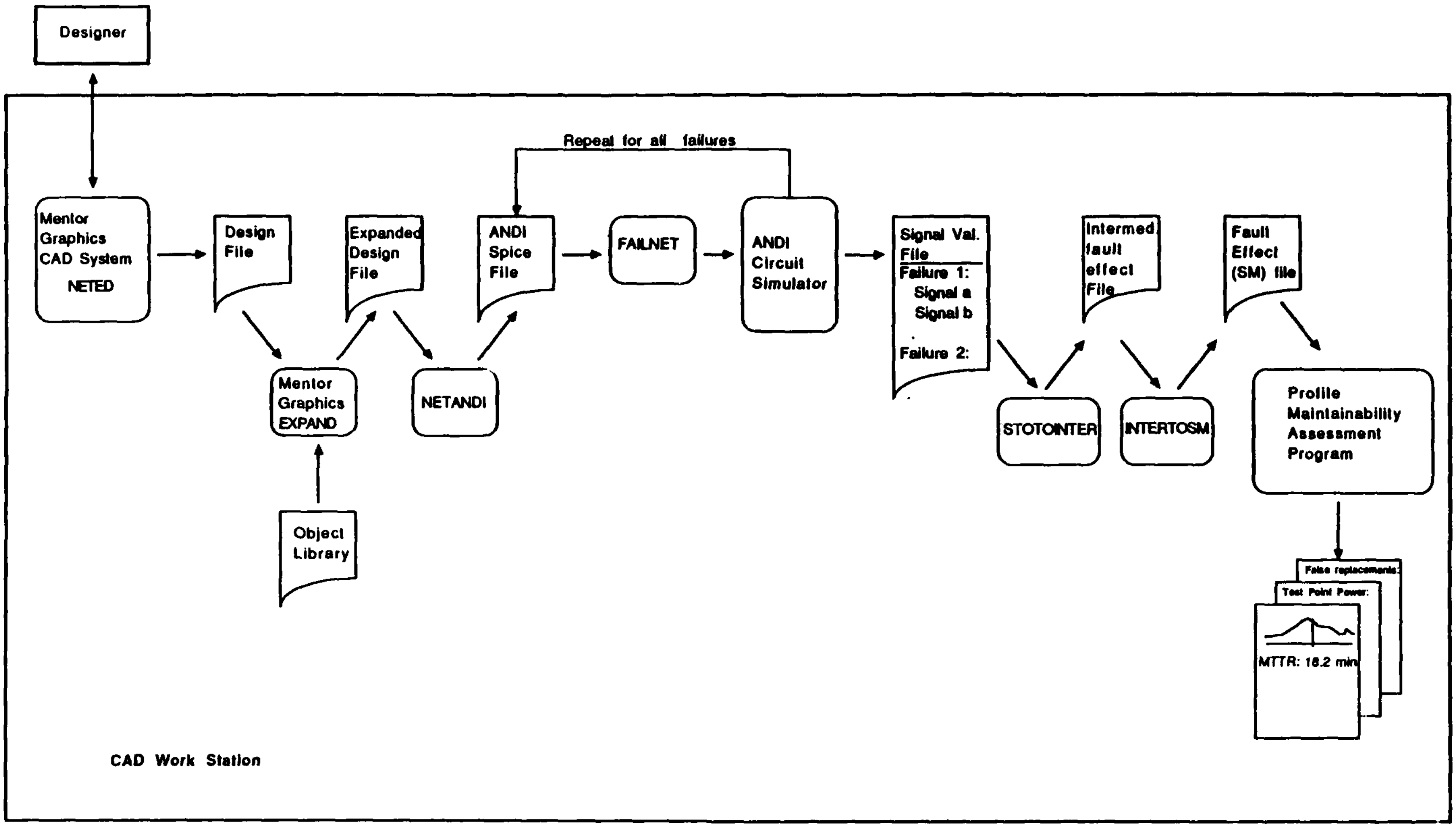


Figure 1. Integrated CAD-Maintainability Assessment System.

model from the netlist, thereby completely destroying the normal function of the component, 2) shorting a digital component's output to ground, thereby emulating an internal failure in an integrated circuit, and 3) applying a 5-volt AC waveform to the output of the component, thereby emulating an internal failure that distorts the circuit output.

STOTOINTER. A program that writes an intermediate data file (called INTERFILE), summarizing the results of the ANDI simulation of each failure. The values of up to 100 test points and indicators are determined by ANDI under each failure condition and written out sequentially to the intermediate data file.

INTERTOSM. A program that reads the intermediate data file, following completion of the ANDI simulation process, and produces a fault-effect data file, expressing normal/abnormal assessments, in Profile format. A key task of this program is to compare the computed values at the test points and indicators with normal values, under each failure condition, and to enter a normal/abnormal classification into the fault-effect table.

Classifying AC and other complex waveforms into normal/abnormal raises interesting questions about the manner in which human technicians perform the function. Of course the human technician cannot even perceive some minor deviations that could be detected via a rigorous quantitative analysis of waveform characteristics. However, even if the technician were to examine the values of the nominal waveform and an observed waveform, he or she would normally accept as normal those deviations that would be expected to arise from one instance of a normal device to another. This judgement in turn requires experience about what extent of variation is likely to occur for various circuit types and component types.

An examination of the test values, as produced by ANDI, indicated that there were very few borderline cases in which a waveform under failed conditions differed just slightly from nominal. This was partly a result of the catastrophic nature of the failures simulated, and partly a natural response of the observed systems to failures. In most cases, for the systems studied, a failure either did not affect an output or it caused major distortions to the normal result.

Consequently, the symptom classification routine employed to interpret the circuit values under various fault conditions was designed to simply detect any differences between the nominal waveform and the actual waveform under failure conditions. If there was any difference, the reading was classified as abnormal. Even though this algorithm is expected to slightly exaggerate the number of abnormal effects, in comparison to judgements made by human technicians, we find that abnormal symptoms constitute a very small fraction of the total available readings. In the SCIACT evaluation, described below, this fraction was under four percent.

Command Scripts. Four "C" shell scripts were developed to automatically call the various programs, and to pass data among them. These scripts perform the identical functions of a user keying in a long sequence of commands at the keyboard of the CAD workstation. While the collection of individual programs and data files comprising the total maintainability analysis system is relatively complex, this complexity is hidden from the user because the analysis process is controlled by the "C" shell scripts. The designer/user only needs to key in the word "Profile", following the creation or modification of the design within the Mentor Graphics CAD system. The scripts then handle the calling of the intermediate programs that introduce failures into the designed system, call ANDI for each failure, write out the resulting fault effects, convert the intermediate data file to a compacted list of normal/abnormal assessments for each failure, and finally call the Profile system to project diagnostic performance.

In addition to the programs and scripts described above, a special object library was created, containing object symbols and underlying specifications compatible with the ANDI circuit simulator. This allows users to create CAD representations in Mentor Graphics' schematic capture system that can be directly simulated with ANDI.

3 Evaluation

Following development of the integrated design environment described above, the newly developed software was installed at Naval Ocean Systems Center (NOSC) in San Diego, California, to operate in association with the Mentor Graphics CAD system already in place. Three designs were then selected for use in evaluating the effectiveness of the analysis system developed:

1. a Doppler Filter circuit, packaged as a single circuit board. This circuit is representative of complex circuits that can be fully designed with computer support, including automated analysis of the circuit behavior.

2. An eight-channel digital signal processor system, consisting of five circuit boards. This system is representative of complex circuitry in which the individual boards are designed and analyzed with CAD support, but are not simulated at the system level due to their extreme complexity.

3. the AN/GSC-40 Satellite Communications Terminal (SCIACT), packaged in eight six-foot racks of equipment. This system is representative of large multi-equipment systems in which system level simulation is infeasible, yet the crucial maintainability design issues are addressed during system integration.

3.1 Application to Doppler Filter Circuit

The Doppler Filter circuit, shown in Appendix A, consists of 34 digital components with a total of approximately 1800 gates, 45 inputs, and 18 outputs.

While there are 34 components involved, there are only seven different types of components represented, five of which are shown in Appendix B. These were added to the Mentor Graphics object library in a form compatible with the ANDI circuit simulator, allowing the design to be entered in Mentor schematic capture and then simulated with ANDI.

For each newly-created symbol, an underlying logic diagram was drawn (Appendix C), using NETED, that expressed the function of the component. The primitive symbols used to construct the components were those understandable to ANDI, viz., resistors, capacitors, transistors, simple logic-gates, and RAM/ROM/PAL digital devices, the symbols for which were previously installed into the Mentor object library. With some limitations the function of most components can be represented as a circuit consisting of these primitive functions. When such a component is used on a schematic, its entire subsystem representation, in terms of ANDI primitives, is transparently substituted in a hierarchical manner.

The library of ANDI primitives was sufficiently rich to allow the logic diagrams found in standard manufacturer data books to be used directly for each component. Upon entering the components to the object library, the schematic of the Doppler Filter circuit was entered, and the analysis process was run.

Operating under script control, the various routines were called to automatically insert failures into the system model, call ANDI to determine the values at the designated test points, write out the symptom data to a file, convert this file into a normal/abnormal form by comparing all readings to normal values, and finally to call Profile for analysis of maintainability.

Limitations of Rigorous Circuit Simulation

A limitation in the ANDI circuit simulator was encountered during initial trials: ANDI provided the capability to monitor outputs at only 100 test points. This capacity is generally acceptable in a design setting, for only a small fraction of a board's outputs are normally designated as test points. Our application of ANDI extended it beyond its normal design purposes. To identify the most informative test points required monitoring all available points and submitting their symptom data to Profile for consideration.

Because of this limitation, the number of possible failures had to be limited to those that could be isolated with 100 test points. In spite of this reduction, *process time to simulate the failures and produce the symptom data for Profile was over twenty four hours*, running on an Apollo DN3000C computer.

This long compute time was a result of our unusual use of the circuit simulator. Conventional (design) use of a fault simulator such as ANDI involves a single execution of the simulator upon the current design, resulting in a file of values and waveforms at each of the test points of interest. Normally, the designer is interested in a limited number of points, and can limit process time by being selective about the values to be monitored. While analysis of a complex circuit might require ten to thirty minutes, that process time does not represent an unacceptable delay in the design mode. Any shortcomings in the circuit outputs are then dealt with via design changes before another analysis is made.

Our use of ANDI, however, involved a complete simulation for each failure in a large sample, combined with the need to track test values at every possible test point in the circuit. Analysis of a single fault therefore could require as long as one hour. Further adding to the process time was the requirement to generate fault effects for most of the failures that *could occur*. While a carefully constructed sample of thirty or forty representative failures might be quite acceptable for estimating MTTR, realistic projections of diagnostic performance for any one of these sample failures is obtained only when the Profile performance model must confront a realistic magnitude of *possible* failures, just as the human technician must. To do this, Profile must have access to the fault effects of most possible failures. For this reason the number of failures that must be simulated far exceeds the sample size required for estimating MTTR.

Development of a Qualitative Approach to Fault Effect Estimation

In some respects the speed with which fault effects are computed is not as crucial as the speed with which Profile analyses are produced. This is so because the fault effects would generally not be affected by changes in the diagnostic features of a system, including rather massive redistributions of circuitry among modules. Thus, even a lengthy fault effect generation process would be tolerable since it could be done one time, prior to performing maintainability studies.

In order to apply Profile to ever larger systems, however, the time to produce the required input data must be kept manageable. Even if extensive CAE analyses are made during the design of a system, the designer is not likely to call for saving signal characteristics at thousands of test points. We cannot expect, therefore, that the required symptom information will be produced during the normal CAE phase. We therefore developed an alternative approach for producing fault effect data for large systems.

A program, GENSM (GENerate Symptom/Malfunction data) was developed to trace signal flows through systems rather than to quantitatively simulate the system behaviors. This qualitative approach operates on the same representation of the design, the design file produced by the CAD system, to determine what components are dependent upon others. GENSM is a relatively straightforward program that operates much like many other signal-flow tracing programs and functional-effect tracing programs. Programs such as LOGMOD (DePaul and Dingle, 1975) and MATGEN (Rigney and Towne, 1977) are similar in their tracing functions. GENSYM was written, however, to operate specifically upon the design file and object library produced with CAD techniques, thereby eliminating any involvement by the designer.

The weakness of inferring fault effects from a topological representation is that internal connectivity of components can change drastically depending upon the particular signal values at their inputs. Since a signal-tracing algorithm does not maintain a quantitative model of the device, the inferences about fault effects are almost certain to involve some error. These same limitations apply to algorithms wherein the *functional* dependencies, rather than physical connectivity, are represented, for the functional relationships among components can change depending upon the quantities of the signals, rendering some of the inferred effects incorrect.

An additional goal was therefore established for the evaluation phase: compare the Profile projections based on fault effects produced by a rigorous simulation of circuit operation to the projections based upon qualitative fault effects inferred from a signal-tracing process.

Results

The signal flow-tracing program was run on the doppler filter circuit, with the same set of thirty failures as were simulated by ANDI. Two approaches were tested: 1) an exact duplication of the conditions simulated by ANDI, i.e., the thirty sample failures were the only possible failures, and 2) analysis of 'all possible' failures, in which every pin of every component was successively failed, yielding 192 possible failures.

The projections for the three analyses are summarized in Table 1.

		Profile-projected repair time *		
	Source of Symptom Data	Minimum	Maximum	MTTR
1	ANDI Simulation, 30 possible failures	50	220	116
2a	BTL Fault-tracer, 30 possible failures	60	150	94
2b	BTL Fault-tracer, 192 possible failures	90	290	136

* Minimum: the time of the failure with least diagnosis and repair time
Maximum: the time of the failure with greatest diagnosis and repair time
MTTR is the Mean Time to Repair for the sample
Assumed that tests require 10 seconds, replacements require 30 seconds.

Table 1. Projected Maintenance Times For the Doppler Filter Circuit

We know that the projections in analysis 1 are more accurate than those of analysis 2a, as the two studies dealt with the same 30 simulated malfunctions and the same set of possible malfunctions, but in study 1 the symptoms were produced by rigorous circuit simulation, whereas study 2a was based upon symptoms derived by signal-tracing. But we also know that case 1 is biased low, as the number of possible failures was limited, thereby producing a simpler problem-solving environment for

Profile. Thus we can conclude that the true MTTR is somewhat greater than 116 seconds. Case 2b, which diagnosed failures from among 192 possible replaceable units,yields an MTTR of 136 seconds, and is believed to be the most accurate of the three. The compute time to produce the fault effects in this manner was just under 16 minutes.

3.2 Evaluation of the Eight-Channel Signal Processor

The eight-Channel Signal Processor is a complete digital system that was under design at NOSC at the time the evaluation of Profile was starting. This system (Appendix D) involved 67 replaceable units (RU's) and 438 possible test points and indicators. It is revealing of the state of CAD technology today (and the complexity of physically compact systems) that this five-board system exceeded the ability of ANDI to simulate its circuitry and generate its failure effects, except in a piecemeal fashion that would not have yielded true system-level fault effects. As a result, only the signal-tracing approach to generating fault effects was used.

The Profile analysis explored one failure in each RU. The distribution of total projected diagnosis and replacement time is shown in Figure 2. The projected MTTR is 462 seconds (assuming bench testing with the test equipment already set up). The four outlying RU's, requiring more than 1000 seconds to isolate and replace were four identical components chained together in a series fashion.

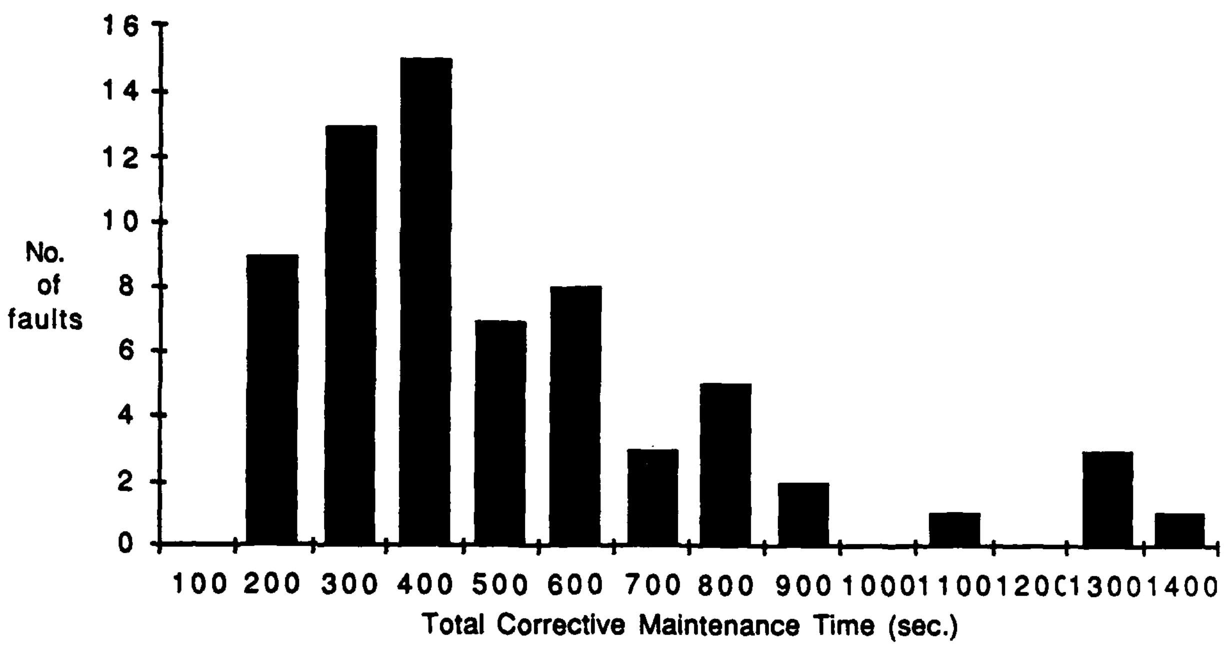


Figure 2. Distribution of Profile-projected Repair Times for Eight-Channel Signal Processor.

Of the 438 testing possibilities, 190 were not used at all by Profile in isolating the 67 sample failures. Not surprisingly, a small number of tests accounted for the majority of Profile's testing, with only 51 test points accounting for 50% of Profile's testing over the 67 failures, as shown in Table 2.

Number of test points or Indicators	Percent of Profile testing time
11	25
51	50
115	75
248	100

Table 2. Allocation of Profile Testing Time to Test Points and Indicators

The detailed Profile analyses identify the particular test points and indicators used by Profile to resolve the sample failures. A designer could consider dropping the unused test points from the design (i.e., not providing physical test points or associated documentation), and could also set out to further reduce the set if desired.

Note that while Profile used 248 test points and indicators to solve the sample failures, *it could have solved the failures with fewer test points, at some increase in MTTR*. Note also that there is no uncertainty about this, because the Profile model never fails to resolve a malfunction. Whenever necessary, Profile resorts to successive replacements (each followed by a confirming test) to completely identify a failure. With extreme deficits of diagnostic features in a design, the Profile solution time projections will soar, as it will have to resort to successive replacement relatively early in the fault diagnosis process for many faults.

The designer's procedure for reducing the complement of test points would be to tentatively eliminate from the design specification some number of test points that were rarely used by Profile, and then to rerun the Profile analysis to determine the MTTR under the design modification. Fortunately, at this stage of use by the designer, the fault-effects are established in a data file, and would not have to be regenerated with each successive maintainability analysis. Any significant functional redesign would, of course, require regeneration of fault effect data.

Upon rerunning Profile, the designer can determine the effect upon MTTR of the tentative design changes, and can decide if the increase in MTTR is a tolerable trade-off for the reduction in production and support costs resulting from the reduced design. Alternatively, a simple command script could be written to repeatedly eliminate one additional test point or indicator from the design, and execute Profile to produce an associated MTTR. The results would provide a clear indication of the most economical selection of diagnostic features.

Projected False Replacements

Because the Profile model operates rationally, it may make replacements of components that are actually good, if those replacements are either necessary because insufficient diagnostic features in the system design prevent complete discrimination of the fault, or they are preferred in terms of the likely payoff compared to further testing. As a result, Profile can project what components are likely to be falsely replaced by a rational technician. In isolating the 67 failures in the eight-Channel Signal Processor, Profile made 39 false replacements, which is a high rate, compared to other system designs.

The replacement rate in this system was affected by the relatively short time required to replace the IC's and the relatively low cost of the components. Both of these factors encourage the strategy of replacing a moderately suspected component as opposed to consuming more testing time. The tradeoff is affected in turn by the cost of maintenance time, in whatever maintenance environment is being evaluated. System restoration time is set to be extremely high when projecting MTTR under combat conditions. This drives the Profile model to essentially disregard the cost of spares, in preference to restoring the system in minimum time. Our earlier research (Towne, Johnson, & Corwin, 1982), indicated a surprising difference in diagnostic strategy under high time cost and low time cost conditions.

For this study, the hourly maintenance cost of depot repair was set at $50 per hour, including facilities costs and other indirect costs. Even if indirect costs are disregarded, so that only the marginal labor cost is applied as the cost of maintenance time, maintenance of digital circuit boards will generally encourage component replacement. Conversely, when Profile simulates the depot maintenance of large systems configured of high-cost replaceable modules, the cost parameters generally drive the model toward minimizing risk of error, prior to replacement. The evaluation of Profile's performance on a large system is described below.

3.3 Evaluation of the AN/GSC-40 Satellite Communications Terminal (SCIACT)

The primary equipment selected as an evaluation vehicle was the AN/GSC-40 satellite communications 'terminal' (SCIACT). This system is configured as eight full-standing racks of equipments (Appendix E) plus such peripheral units as operator console, keyboard, printer, and disk drive. Normally racks one through five are installed in a separate area from racks six through eight.

The considerations that argued strongly in favor of SCIACT as an evaluation unit included these:

- the system had been fully redesigned at NOSC, and was being supported at NOSC, so technical documentation and expertise was available.

- a formal Maintainability Demonstration Test of the unit had been conducted at NOSC in 1982, involving the timed diagnosis and repair of 46 inserted failures (five failures in the computer could not be used in the evaluation due to changes in the maintenance policy made subsequent to the Test). The time data for this study were available (NOSC Report SCA-CP-00240, 1982) for comparison to Profile projections for the same failures.

The organizational maintenance policy for the system calls primarily for substitution of modules, although some narrow-band units are repaired via circuit board replacement. In all, there are 102 replaceable units.

The SCIACT system includes a built-in-test (BIT) capability that exercises various operating modes of the system and returns diagnostic information to the maintainer about the status of various subsystems. The maintainer then pursues a self-directed testing strategy, although the technical manual provides some testing procedures as guides. Virtually all of the 102 replaceable units can be isolated from front-panel indications. These indications include the BIT results, manually controlled front-panel indications, and indications obtained by communicating or not communicating with other stations. In all, the SCIACT design provides 83 such indicators of system operation.

SCIACT is representative of larger systems consisting of individual units that may be designed with CAD techniques, but would not permit simulation of system-level operation under today's CAD/CAE technology. It is such systems as SCIACT, however, that present the greatest challenge and opportunity for affecting maintainability, for the designer has numerous packaging and diagnostic design options at this level, and the system complexity challenges the designer's capacity to reliably and quantitatively evaluate the impact of various design options. *A basic objective of the SCIACT application was therefore to determine the most effective manner of capturing the features of the system design needed to analyze maintainability.*

While rigorous circuit simulation was clearly infeasible, we considered the possibility of generating fault effects using the signal-tracing approach used for the eight-channel Signal Processor. Such an approach appeared feasible since the number of replaceable units was quite manageable as were the number of major signals among these units and the number of test indicators.

Our attempts to produce correct fault effect data based upon dependencies among the system units, either at a functional or physical level, did not produce sufficiently accurate data, however. The internal complexity of the replaceable units defeated all attempts to treat them as components, for their internal connectivity changed radically from one operating condition to another.

The only feasible approach therefore was to enter the abnormal effects of each failed replaceable unit directly into the Profile fault-effect table. While the number of possible symptoms in this table is large (102 replaceable units x 83 indicators = 8,466 symptom cells) the actual number of abnormal effects for each replaceable unit was quite limited.

Determining the abnormal tests for each failure condition required approximately two man-weeks of effort, following a period of SCIACT familiarization that would not be required by the designer of a system. This work was done by a NOSC designer (Mike Dwyer) and one of the authors (Mark Johnson), neither of whom were as knowledgeable of SCIACT's functional operation as the original designers would be (Dwyer was fully knowledgeable of the physical packaging of SCIACT, as he had played a key role in designing that aspect of SCIACT).

The completed fault effect data consisted of only 333 abnormal effects, or just under four percent of the possible test results. Only two power supplies and a disk drive produced abnormals at more than four indicators. The reasons for this are 1) SCIACT is a multi-mode system with most replaceable units devoted to a subset of operational modes, and 2) some of the indicators, such as the BIT, displayed extremely rich information, thus the *number* of abnormal indicators was deceptively low. The BIT display alone, pointed the technician to a failed subsystem. This richness of display information was evident in the fault effect data presented to Profile. Some indicators had as many as six possible symptom characteristics.

The remaining data required by Profile consisted of 1) time data for the possible tests and disassembly/assembly operations, 2) cost and reliability data for the replaceable units, and 3) conditional system state information.

Task Times

Profile evaluates the expected benefit of each alternative diagnostic action and the time cost to perform it, and selects that course of action yielding the highest expected return. To do this, Profile requires the time to disassemble down to each unit, replace it, and then reassemble the system. The total time for each unit is determined by summing the times to remove and replace all the parts that must be removed to gain access to the unit.

These basic times are extracted from a library of basic maintenance times. If Profile should come into general use, it is expected that users would share and expand such a library of task times. For the study of SCIACT, times were generated for those tasks not previously analyzed for earlier Profile applications. The times for the newly added tasks were produced by adding up the times to perform each motion required. The basic motion times were retrieved from a standard industrial engineering motion time base, called Methods Time Measurement (MTM). A sample analysis of one task is shown in Appendix F. Appendix G lists all the basic maintenance task times used to quantify SCIACT operations.

The total SCIACT unit replacement times were determined by summing the basic removal and replacement times of all the components restricting access to each unit, as shown in Appendix H. Because the design employed highly standardized packaging throughout, there were relatively few kinds of different screws, access doors, cables, and circuit board securing devices. The unit replacement times were used by Profile in partial consideration of the rational next step, and to produce total fault isolation and repair times for the 46 sample failures.

Cost and Reliability

The relative Mean Time Between Failures (MTBF) was estimated for each replaceable unit and entered into the data base. These figures allow Profile to concentrate its efforts initially on the less reliable sections of the system, progressing to highly reliable areas only when symptom information indicates that to be wise. Since the failure rate values are *relative*, it is only necessary to allocate failure likelihood reasonably well.

For this study all RU's were assigned equal cost. This was done in an effort to reproduce as closely as possible the conditions of the Maintainability Demonstration Test, in which the technician had access to all necessary spares, and was not likely to be affected by component cost. This would not be the case in a true depot situation, in which replacement of costly units would be avoided if possible.

Conditional System State Data

Because Profile maintains an internal model of the condition of the physical system as Profile simulates the performance of actions that would change the state of the system, it is able to make all time estimates sensitive to actions already performed. Thus a test that might be time consuming at one point in a diagnostic sequence might be readily performed at some other point, after some of the prerequisite actions had been accomplished to meet other goals.

A review of the SCIACT architecture revealed no significant cases wherein test times changed as a result of partial disassembly, i.e., all tests were done at the front panel. Because of SCIACT's physical expanse, however, the time to accomplish a test was greatly affected by the location of the maintainer following a previous test. To represent this situation, each test was assigned a state number, corresponding to the rack at which it is performed. A simple transition table was then entered, in the standard Profile format for conditional times, that expressed the time to transition from each state (rack) to each other state (rack). When Profile considered the time to perform each test, then, it recognized the time advantage of remaining at the current rack, and weighed this against the time cost and diagnostic value of walking to another rack.

In the most detailed format of analysis output, Profile lists the state changes it would produce to follow one test with another. In this application, therefore, Profile indicated that the simulated maintainer would walk from one rack to another, whenever that state change was found to be worth the time investment.

Results

Computer process time to analyze the 46-failure sample was 52 minutes. This was entirely Profile compute time, as no automated fault generation was required.

Table 3 compares the Profile projections to the actual diagnosis and repair times for the 46 failures used in the Maintainability Demonstration Test.

	Actual	Profile
Minimum	2.7	4.2
Maximum	36.7	17.4
Mean	10.7	10.5
Std. Dev.	6.4	3.5

Table 3. Actual and Projected Maintenance Times for SCIACT (min.).

As shown in Figure 3, below, the distribution of Profile projections corresponds generally well with the distribution of actual times, except that Profile predicts more repair times in the range of 14 to 16 minutes, and none over 18 minutes.

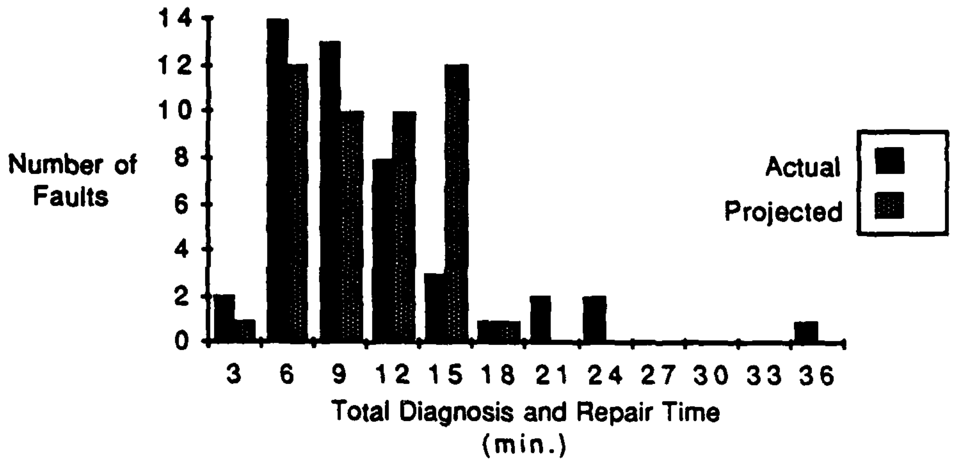


Figure 3. Actual and Projected Repair Times: SCIACT

The results obtained here are very similar to those obtained previously, viz., the Profile MTTR projection corresponds very closely with the mean of actual times, but the variation in the Profile distribution is significantly less than actuals.

Fortunately, the documentation accompanying the Maintainability Demonstration Test noted when unusual circumstances occurred during the repairs. On 8 of the 46 failures there was either a significant technician error in performing tests, or a significant shortcoming in either tooling or technical documentation (subsequently resolved) that affected normal diagnosis and/or repair. While it would be extremely beneficial if Profile could detect and quantify such shortcomings in the maintenance resources brought to bear, its projections pertain to correct testing procedures carried out in an environment of adequate tooling and documentation.

If the eight noted failures are omitted from the SCIACT Maintainability Demonstration Test data, the standard deviation drops to 4.4. Thus Profile's standard deviation estimate of 3.5 should be viewed as a lower limit estimate of variation, when all unusual circumstances are eliminated. Earlier controlled applications of Profile show a closer correspondence between the projected variation and actual, partly because experimental conditions assure adequate tooling and documentation.

Profile's projections concerning relative difficulty of problems corresponded well with the actual distribution. The failure requiring minimum time for the human technicians (2.70 minutes) was also the failure for which Profile predicted minimum time. The three failures requiring greatest actual time (36.7 minutes, 24.3 minutes, and 23.4 minutes respectively) all involved technician error or lack of a needed tool. The failure requiring greatest time without unusual problems required 20.6 minutes, which is reasonably close to Profile's estimate of 17.4 for the most difficult problem.

Ease of Use

The technical/clerical effort required to apply Profile to SCIACT was surprisingly manageable, in spite of the size of the system. Although the SCIACT application did not permit the fully automatic CAD-to-Profile process, the fault effect data were prepared in less than two man-weeks, once a level of technical understanding had been attained. Because the fault effects were produced in this manner, the replaceable units did not have to be decomposed into many smaller functions to support automated circuit simulation. As a result, SCIACT could be represented in terms of its 102 replaceable units, making it a relatively simple system to describe.

Several significant simplifications were made to the Profile data base format as a result of this evaluation. One of these eliminated a large amount of redundant naming of replaceable units in the fault effect table; another eliminated redundant symptom specifications associated with the failures to be used in the sample. In addition, the designer using Profile recommended development of a simple editor for entering symptom data in those cases in which the Profile database is not produced automatically.

While a detailed user's manual was not available, the NOSC designer acquired an understanding of preparing Profile design specifications and running Profile following two part-day demonstrations.

4 Summary and Conclusions

This technical report has summarized the final phase of development of the Profile technique: an on-site installation and application of the computer software in an environment as close to the design setting as possible. Because a major objective of the work was to evaluate the accuracy of Profile projections, it was important to select as one evaluation vehicle an equipment for which some maintenance history was available. The AN/GSC-40 Satellite Communications Terminal (SCIACT) was selected because it was large, complex, and offered a set of unusually precise corrective maintenance time data.

Other objectives had to do with demonstrating and testing the degree of automaticity that could be achieved in passing a completed design specification from a commercial CAD system to Profile, and with the feasibility of generating fault effects based upon a qualitative analysis of system architecture rather than upon quantitative computation of circuit values. Smaller, but complex, circuit boards were selected as the vehicles for conducting these phases of the evaluation.

4.1 Summary

The results of the evaluation have been very encouraging in the following ways:

1. Designs of circuits produced using CAD techniques were analyzed entirely automatically by Profile, in response to a single command entered at the keyboard of the CAD workstation.

2. Fault effects generated by qualitative analysis of system architecture produced maintainability projections that conformed reasonably closely with those based upon rigorous, quantitative analysis, thereby reducing the compute time immensely.

3. A very large system was analyzed with a modest one-time data-preparation effort. The projected MTTR was extremely close to the mean time of the 46 sample failures previously timed.

4. The skills required to apply the Profile technique were easily taught. A NOSC designer was able to independently apply Profile with a minimum of instruction.

There were some negative findings as well, however:

1. The compute time for Profile to analyze the large SCIACT system, following production of the required fault-effect data, was just over one minute per sample

failure, when the total set of possible failures was 102 replaceable units. This compute time, while tolerable for evaluating the impact of major design alternatives, would not allow truly interactive cooperation between the designer and the machine. This compute load also discourages certain promising and fully automated approaches for identifying optimal test point selections, through analysis of a large number of candidate sets.

2. The projected variability about the mean maintenance time was significantly less than the actual. While much of the error in projecting variability can be explained in terms of technician errors and shortcomings in the maintenance environment at the time the Maintainability Demonstration Test was performed, such difficulties are certain to arise in the actual maintenance environment as well. We currently lack the ability to predict with reasonable accuracy the likelihood of the various errors that a typical technician might commit.

3. The automatic link that was developed between the CAD process and Profile maintainability analysis is currently limited by the compute speed and simulation capacity of the available commercial CAE software. The use of *quantitative* circuit simulation will be limited until 1) the compute speed is increased by a factor of approximately 100, and 2) the test point capacity is increased by a factor of approximately 10.

4.2 Conclusions

It is important in such a final report as this to attempt to accurately characterize the significance of the achievements and the seriousness of the remaining problems. This work has concentrated on determining the practicability of analyzing maintainability of systems based upon their design specifications, and doing so during the design phase. The utility programs and procedures developed in this final phase of the development program had as their goal to effectively link existing analytic processes together in a cohesive and useable manner. As such the conclusions pertain to the success with which the combined systems performed, and the problems that emerged.

Perhaps the most significant capability demonstrated by this study was that of producing a maintainability analysis directly and automatically at the point the designer has produced a CAD representation of the system design. Correspondingly, the greatest disappointment is that current commercial CAD/CAE tools exhibit power and capacity limitations that restrict the domain of application. The crucial maintainability questions do not arise until the system under design involves many circuits being combined in many boards, modules, and equipments, yet it is at that point that the commercial CAE tools reach their limits.

Tractable Problems

The test point capacity limitation we encountered is not expected to be a major problem for the future. The 100 test point limit in ANDI is relatively arbitrary, and was probably established by considering the upper limits that would be required for a conventional analysis of circuit behavior. High capacity disk drives are certainly capable of storing the signal characteristics at thousands of test points. It is reasonable to expect that a 100-fold increase in the capacity of circuit simulators could be achieved any time the commercial CAE vendors choose to implement such changes. It will require some pressure from applicators, however, to communicate this need to the CAE developers

The compute-time limitation is also not of a magnitude that is particularly worrisome. Ten-fold to hundred-fold increases in compute speed are generally achieved every two to four years. While we would expect system complexity to also increase during this time, there are some promising approaches for achieving several orders of magnitude decreases in compute time that are related to the compute process, rather than the inherent speed of computation.

While the huge compute load to produce precise fault effects was startling, this process does not represent a serious obstacle to performing maintainability analysis. The qualitative signal tracing approach developed during the evaluation reduced compute time to about one percent of the former time, for this stage of the maintainability analysis. Even if fully precise fault effects are required in some applications, they need only be produced one time, unless functional changes are also made to the design. Furthermore, we could expect that the signal values would have already been computed during the design of the functional system, if it was done using CAD/CAE techniques of the near future.

Thus the only remaining obstacle to truly interactive maintainability analysis is the Profile system itself. There appears to be at least one way to revise the Profile analysis function to execute considerably faster. This promising approach would have Profile store intermediate results (suspicion levels, hypotheses, states of a partially disassembled system, etc.) following each test selection for a problem. After completing one sample fault, Profile would have built one branch of a very large decision tree.

Upon starting to analyze the second fault in the sample, Profile would not have to compute the first test; it would be the same selection for all faults (given that the system starts with the same initial information). Furthermore, to the extent that the symptoms of the current fault match those of one previously analyzed, deeper nodes in the decision tree would apply without making the time-consuming determination of what test to perform next. Thus, as each succeeding fault is analyzed, and branches are added to the tree, the compute time to explore additional faults would diminish drastically.

This design of a faster compute algorithm has no bearing upon the diagnostic model that is employed in Profile. It is simply a promising way to restructure the execution of the model so that a large set of problems can be modeled in an efficient manner. The current approach was implemented primarily because of its direct reflection of the diagnostic model, and because of the ease with which it permitted tracing and revising the diagnostic reasoning applied therein.

Serious Problems

The difficult problems emerging from this work have to do with foreseeing and understanding human error, so that realistic projections can be made in non-idealized maintenance settings and good estimates of variability can be produced. The crux of the difficulty lies with expressing error commission mechanisms in operational terms, such as a mismatch between the requirements of a task and the abilities of the individual technician. The current state of understanding and predicting human performance does not come close to what is required to work at this level. The only practical recourse is to employ gross error likelihood rates based upon some form of generic task taxonomy. Such an approach is in itself of immense proportion, and would contribute little to the basic understanding needed to progress to a more enlightened and fundamental level.

The second part of dealing with human error has to do with predicting consequences of error, and here the existing Profile model appears to contribute much that is needed. There is considerable potential for introducing errors into 1) the Profile fault effect belief structure, to represent misconceptions about how the device operates normally and in various failed states, and 2) into the test performance section of the model, to represent errors in conducting diagnostic operations.

Errors in Beliefs. During early experimentation with alternative diagnostic models, it was found that the diagnostic performance of well-qualified but non-perfect technicians could be closely approximated by an efficient diagnostic strategy operating upon quite imprecise, but error-free, fault-effect data. Here, an error would be believing that a particular fault could not affect a particular test result, when in reality it could, or believing that a particular fault could affect a particular test result, when in reality it could not.

The most realistic testing performance by Profile was obtained when the symptom data for each fault accurately reflected the possibility of an abnormal symptom for every test that actually would be affected by that fault, but provided no quantitative information about the nature of the abnormal reading. Thus the diagnostic model was prevented from quickly converging upon a fault by detecting that only a very limited set of possibilities could have produced the exact value observed.

The fault effect data structure also provides a simple way to express uncertainty about a fault effect. This was found to be crucial, for even the designer may be quite uncertain about whether a particular failure will be exhibited at a particular test point. Such uncertainty stems from two sources. First, the complexity of today's digital circuits exceed the capacity of human beings to mentally simulate their behavior in either normal or failed conditions. Secondly, uncertainty stems from not being certain of the exact nature of the failure (for example, has a pulse circuit shifted by 11,000 cycles per second or by 11,005) and the precise values of all the system components, which may vary slightly from one unit to another Thus, even after performing a rigorous CAE analysis of a circuit, there may always remain uncertainty about some fault effects in fielded units.

Misconceptions and uncertainties of an individual technician (or a hypothesized representative technician) can be represented by perturbing the fault effect data to match those erroneous or unestablished beliefs. We would expect the Profile model to rather accurately predict the first few tests performed by a technician holding those flawed or missing beliefs, but we would also expect serious departures between true performance and Profile projections as the problem progresses, for the human technician has the capacity to constantly revise his or her belief structure, while Profile currently does not.

We have observed the performance of many hundreds of real-world troubleshooting problems, the great majority of which involved some degree of error. Usually, the technician receives some cues that something is wrong. Either test values don't appear to be providing a consistent body of evidence, or there is no imaginable fault that could be producing the observed symptoms. Adding to the technician's problem is that he or she may suspect that one or more of the test results obtained were affected by errors in performing the test or in recalling its result. Thus the technician may be maintaining multiple levels of hypotheses, not only concerning the state of the unit under test, but also meta-hypotheses about what he or she may have done incorrectly. It appears, therefore, that the fault-effect data structure might represent many erroneous beliefs, and uncertainty, but the process for modifying those conceptions during and between fault diagnosis experiences is beyond our current understanding.

In a similar fashion it appears quite feasible to project the performance of individual technicians holding well-defined misconceptions about testing procedures, if those misconceptions could be known. To do this would simply involve providing to Profile's test interpretation routine the test result that would be obtained if it were performed according to the technician's flawed understanding of the procedure. Profile would then make a rational evaluation of the test result and proceed. Of course, this incorrect test result would *usually* extend the number of tests required to resolve the fault, and we would expect to obtain a relatively accurate projection of the impact of that performance error, as long as it persists.

Here, the Profile model lacks a mechanism for hypothesizing that there are errors in the test results themselves, and in ultimately making corrections to low-level test procedures responsible for making those tests. In fact, Profile's symptom information is simply retrieved from the fault effect data, i.e., it is always correct. While errors could be introduced, there is currently no way to do so based upon some theorized misconception in testing procedures, although this does not represent a serious theoretical problem. More seriously, we have little understanding how the human technician proceeds to correct flawed procedures based upon evidence obtained during his or her diagnostic performance.

Most of this technical report has been devoted to a field validation study of one particular maintainability method. As is evident from the earlier sections, the method proved to be a good predictor of actual maintenance behavior, when applied to real technicians working on real equipments. This is especially gratifying when we consider the great apparent variety of performance that is exhibited by diagnosticians as they opportunistically respond to the particular situations established by each unique failure in each unique device.

References

DePaul, Ralph A. Jr., and Dingle, Larry A. 1975. LOGMOD Diagnostic Procedures for Mechanical and Electrical Equipment, ASRO Report, Moffett Field, California.

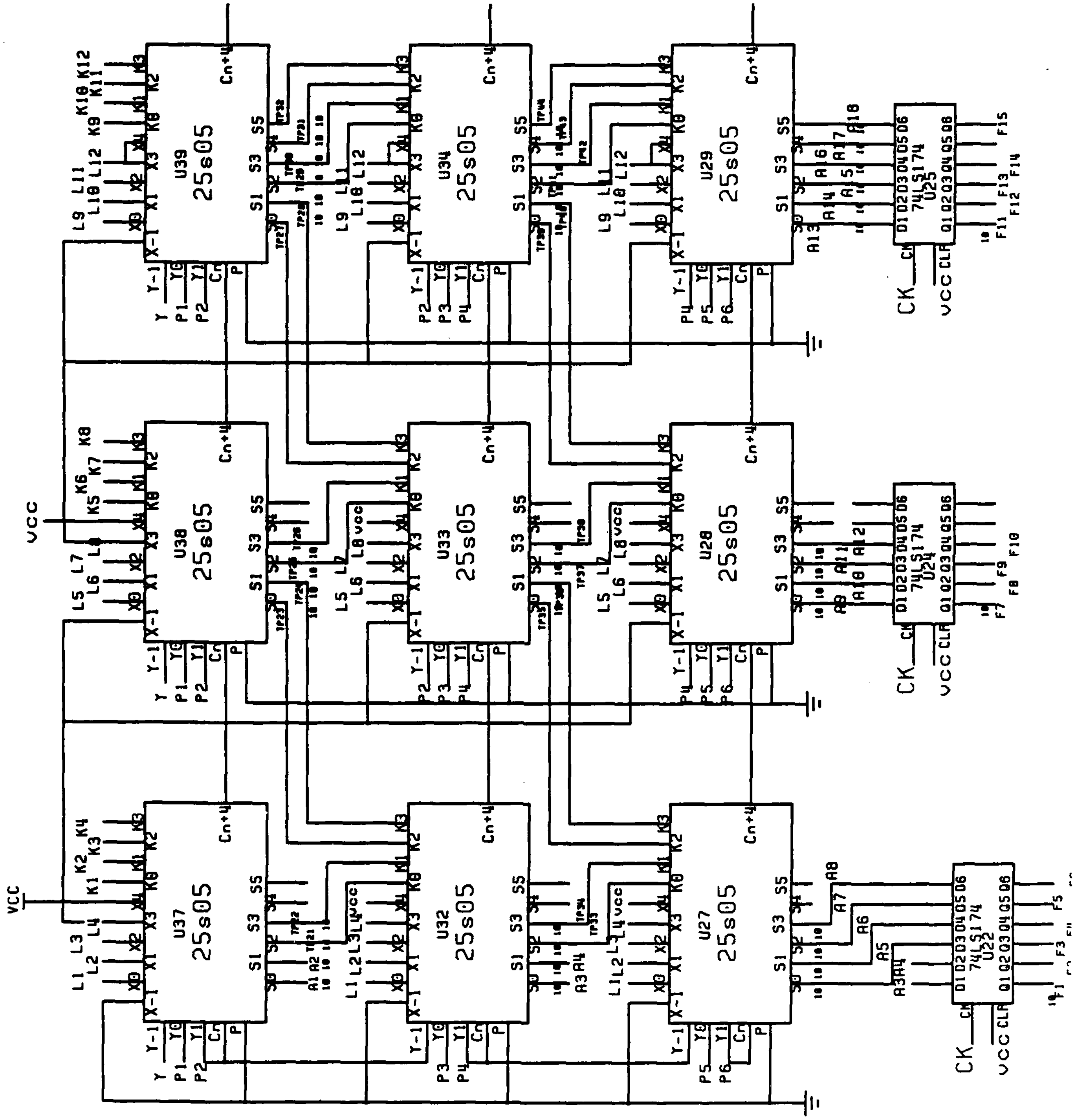
Rigney, J. W., and Towne, D. M. 1977. Some Concepts, Devices, and Procedures for Improving Technical Training and on-the-job Performance of Maintenance. Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, Report NAVTRAEQUIPCEN 76-C-0023-1.

Towne, D. M., and Johnson, M.C., and Corwin, W.H. 1982. PROFILE: A Technique for Projecting Maintenance Performance from Design Characteristics. Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, Report 100.

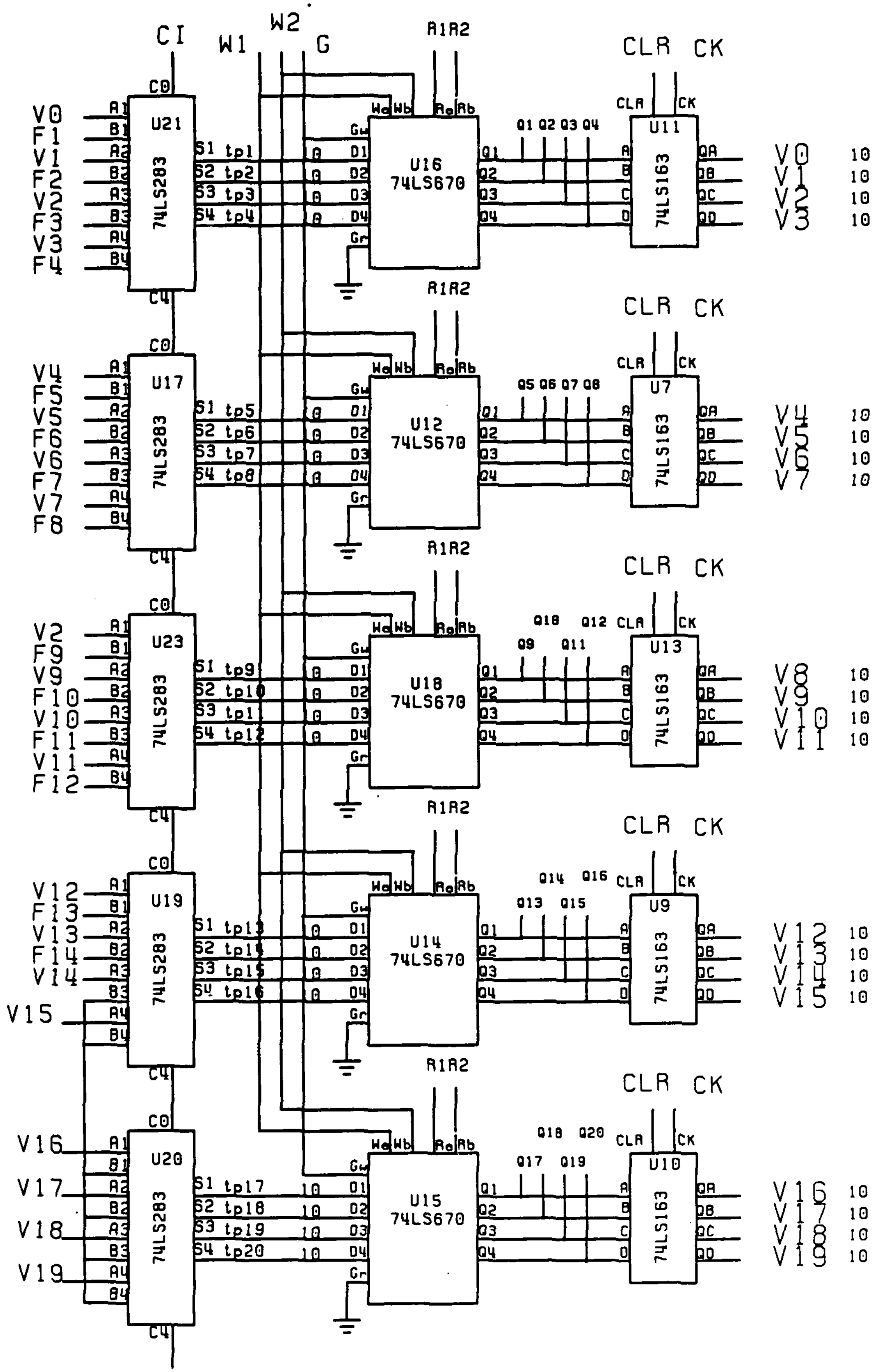
Towne, D. M., and Johnson, M.C. 1987. Research on Computer Aided Design for Maintainability. Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, Report ONR-109.

Veisel, S.K. and Dargis, S.W. 1982. Satellite Communications Terminal AN/GSC-40 Maintainability Demonstration Test Report. San Diego, CA: Naval Ocean Systems Center, Report SCA-CP-00240.

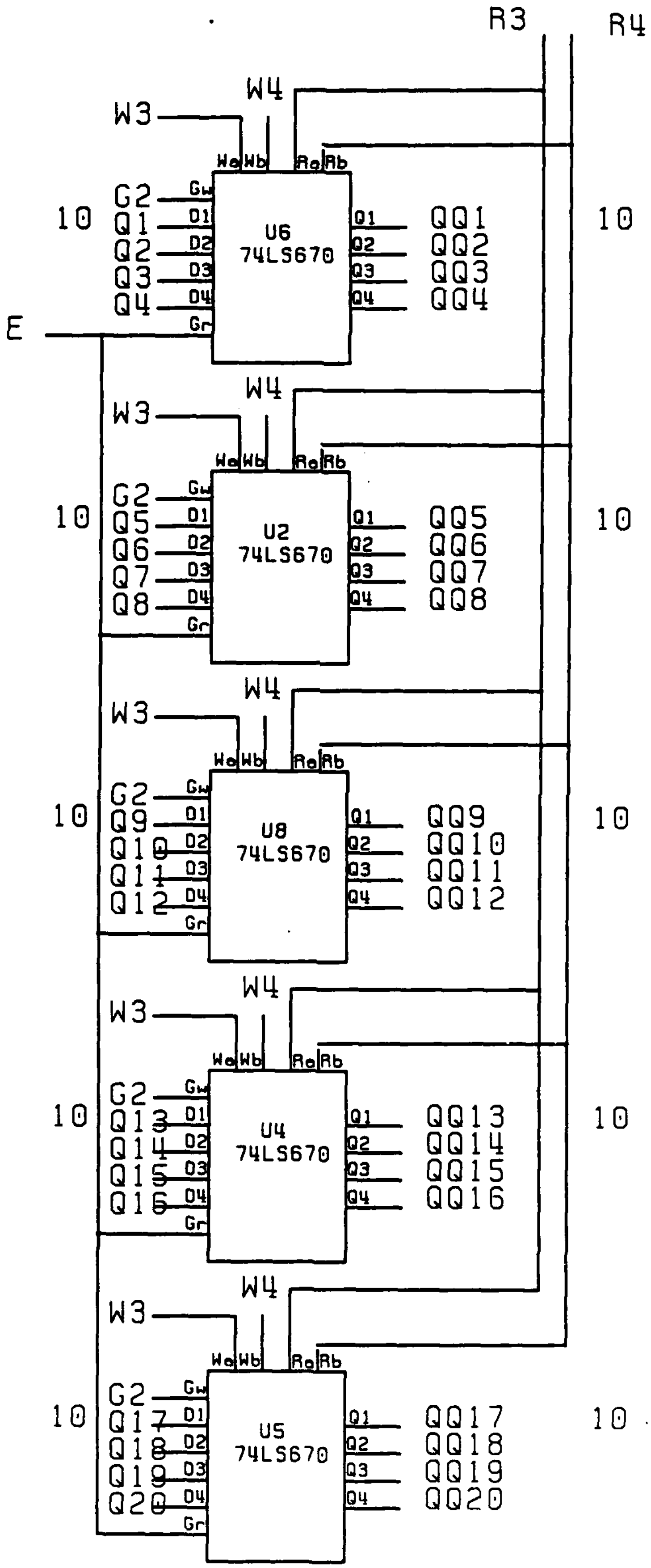
Appendix A - Doppler Filter Circuit
(screen print of CAD representation)



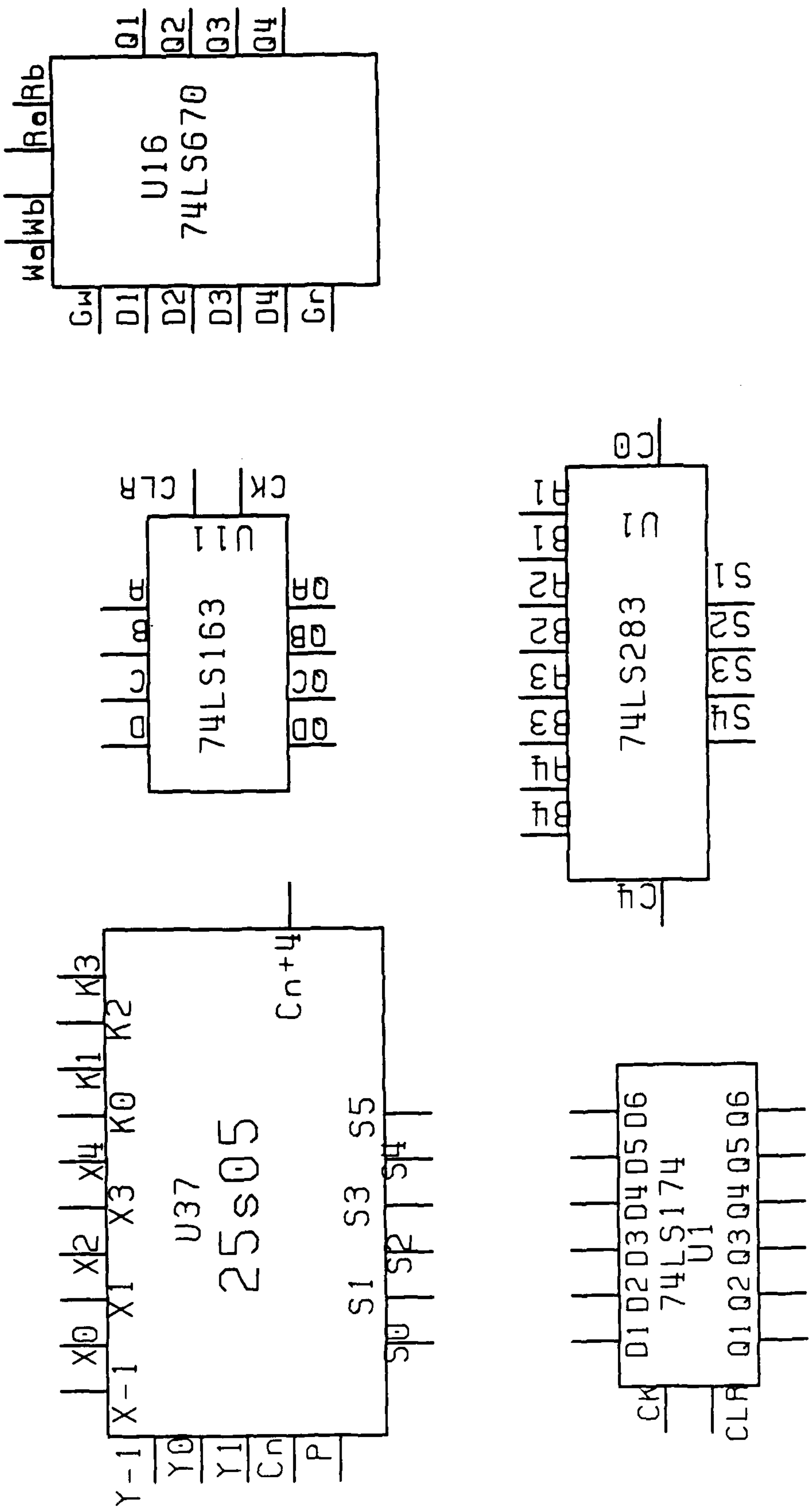
Appendix A


CI
W1
W2
G
R1R2
CLR
CK
U21
74LS283
U17
U23
U19
U20
U16
74LS670
U12
U18
U14
U15
U11
74LS163
U7
U13
U9
U10
V0
F1
V1
F2
V2
F3
V3
F4
V4
F5
V5
F6
V6
F7
V7
F8
F9
V9
F10
V10
F11
V11
F12
V12
F13
V13
F14
V14
V15
V16
V17
V18
V19
S1 tp1
S2 tp2
S3 tp3
S4 tp4
S1 tp5
S2 tp6
S3 tp7
S4 tp8
S1 tp9
S2 tp10
S3 tp11
S4 tp12
S1 tp13
S2 tp14
S3 tp15
S4 tp16
S1 tp17
S2 tp18
S3 tp19
S4 tp20
Q1 Q2 Q3 Q4
Q5 Q6 Q7 Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
Q16
Q17
Q18
Q19
Q20
V8
10

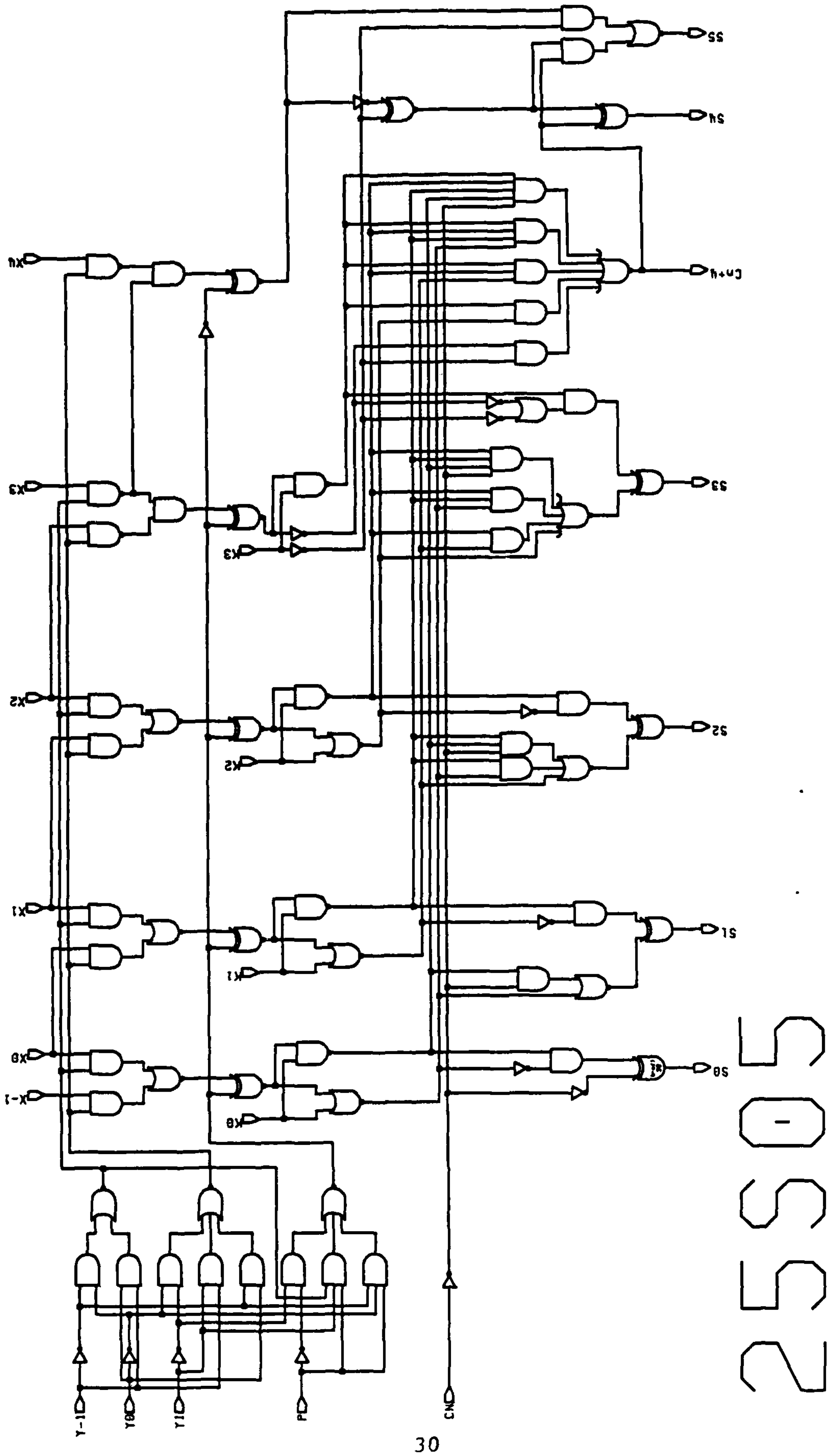
Appendix A


R3
R4
W3
W4
Wa Wb Ra Rb
G2 Q1 Q2 Q3 Q4
Gw D1 D2 D3 D4 Gr
U6
74LS670
Q1 Q2 Q3 Q4
QQ1 QQ2 QQ3 QQ4
10
10
E
U2
74LS670
G2 Q5 Q6 Q7 Q8
QQ5 QQ6 QQ7 QQ8
U8
74LS670
G2 Q9 Q10 Q11 Q12
QQ9 QQ10 QQ11 QQ12
U4
74LS670
G2 Q13 Q14 Q15 Q16
QQ13 QQ14 QQ15 QQ16
U5
74LS670
G2 Q17 Q18 Q19 Q20
QQ17 QQ18 QQ19 QQ20

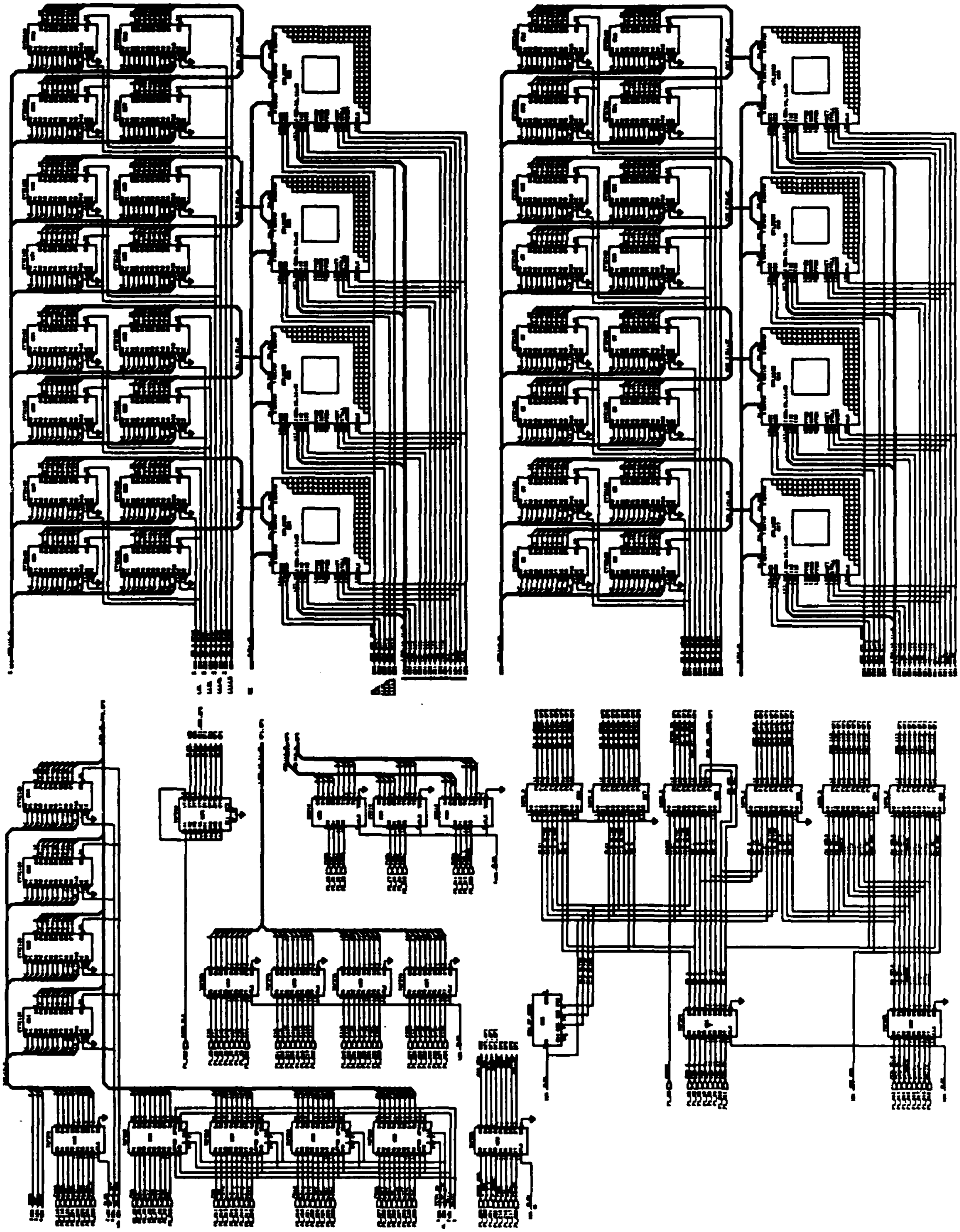
Appendix B - Portion of ANDI Object Library (Doppler Filter components)

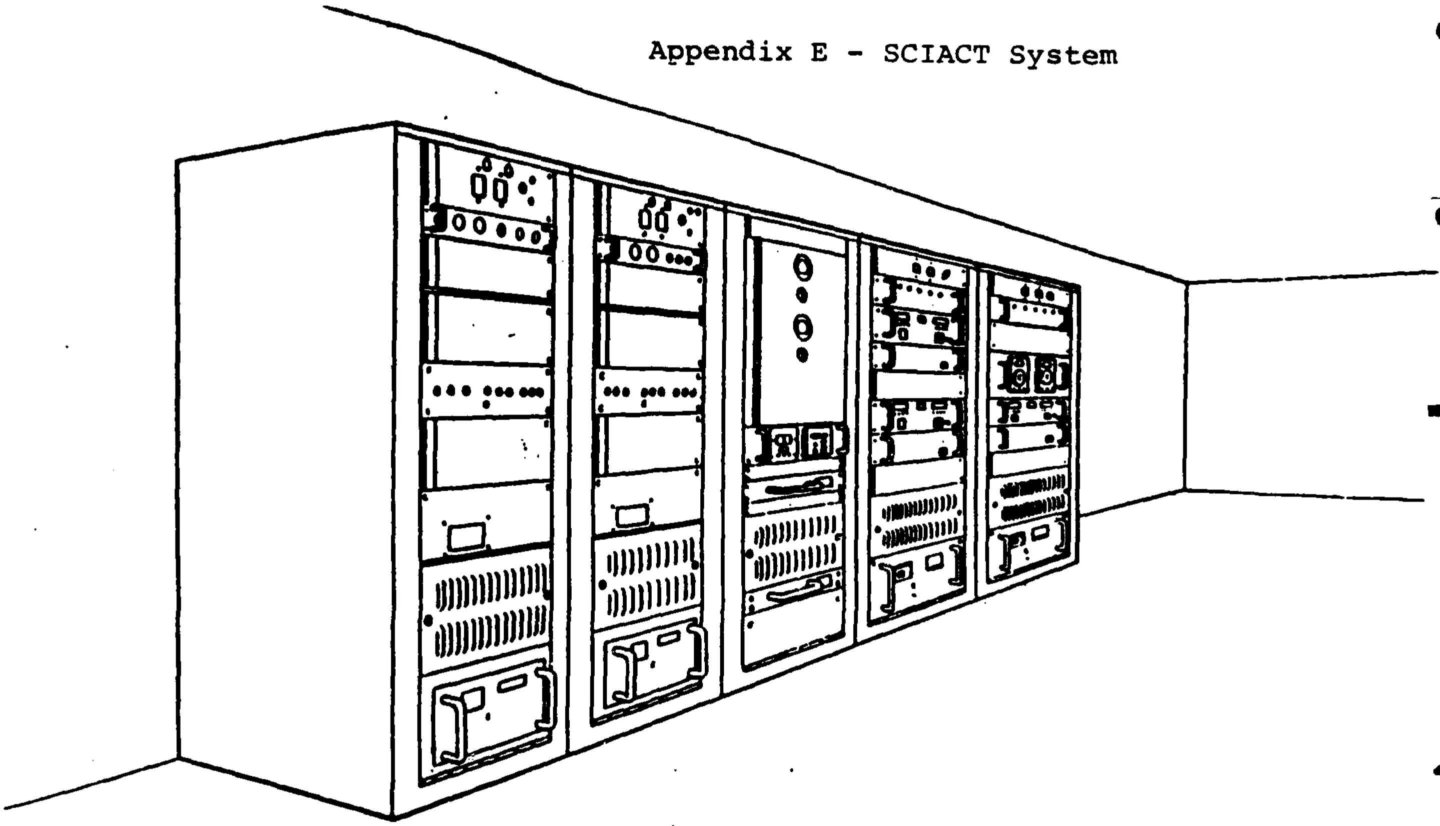


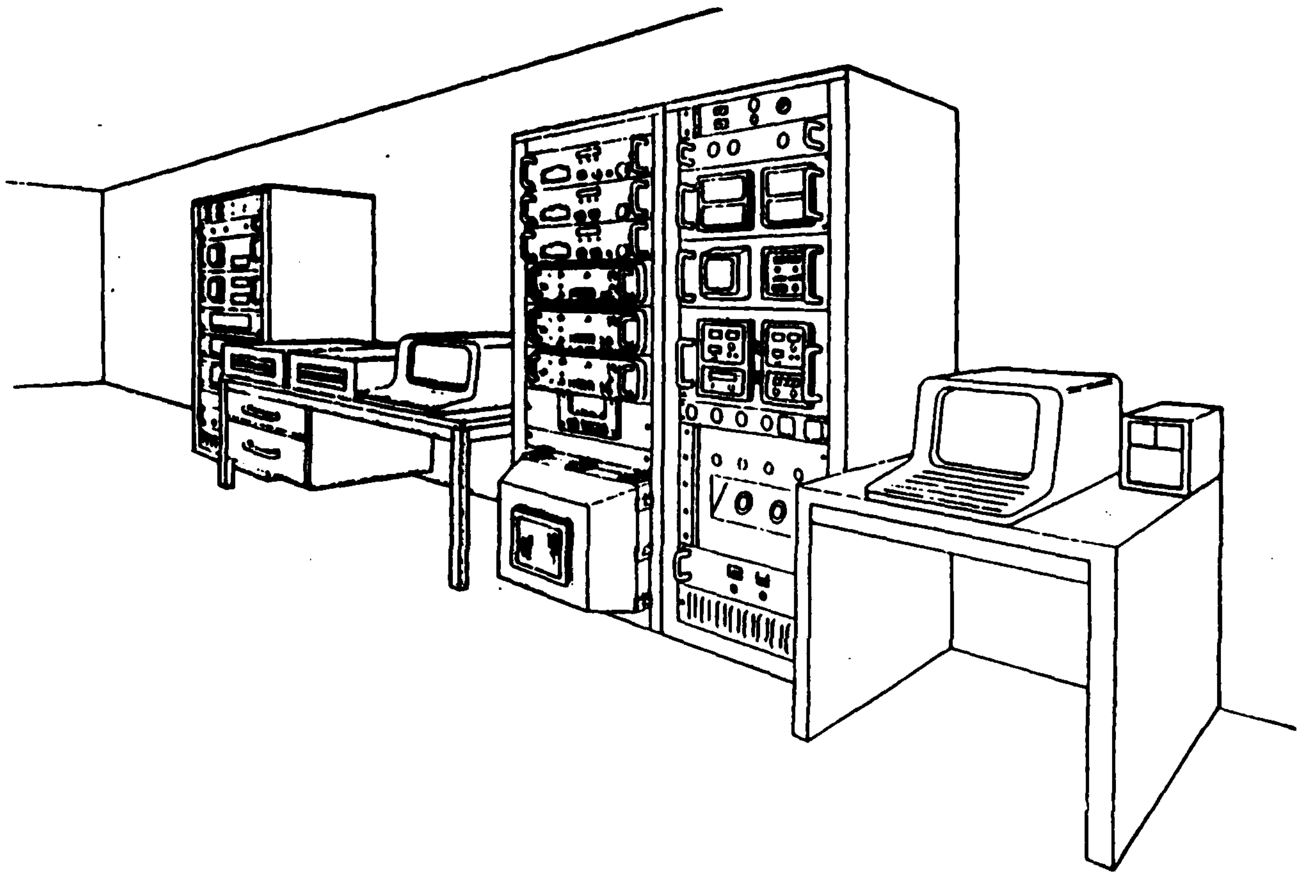
Appendix C - CAD Specification of a digital component



Appendix D - Eight-channel signal processor







Satellite Communications Terminal AN/GSC-40,
Typical Operation and Command Control Center

Detailed Analysis

9/22/88

Element Code: ANUT
Element Name: Assemble nut & washer w/wrench
Total Time: 977

Description	Code	Freq	Time
Move washer to bolt	18m	1	18
Position washer on bolt	9p	1	45
Slide washer onto bolt	4.20m	1	98
alter grasp to hold washer	2.2g	1	12
Move nut to bolt	18m	1	18
Position nut on bolt	10p	1	50
Finger move to start thread	1f	2	6
Move to turn down	2m	20	128
Get to turn down	2.2g	20	241
Get wrench	12.2g	1	20
Move wrench to nut	18m	1	18
Adjust size	1f	5	16
Move wrench away and back to nut	8m	2	21
Position wrench on nut	10p	2	100
Position wrench flush to washer	3p	1	15
Move first time	4.20m	1	98
move back from first turn	8.2g	1	17
Move to tighten	4m	1	8
Apply pressure to tighten	ap	2	30
Move wrench away	18m	1	18

Appendix G - Basic Maintenance Task Times

Basic Maintenance Task Times (min.)

Remove/replace:	**Remove**	**Replace**	**Total**
cable release	0.11	0.11	0.22
circuit board	0.19	0.11	0.30
connector, quick-release	0.05	0.29	0.33
connector, threaded	0.25	0.44	0.69
door, hinged	0.09	0.03	0.12
fastener, quick-release	0.06	0.06	0.12
module	0.65	0.90	1.55
nut with washer	0.48	0.58	1.06
nut, knurled	0.12	0.11	0.23
retainer with nut	0.11	0.11	0.22
screw, knurled	0.12	0.11	0.23
Screw, Phillips-head	0.14	0.18	0.32
side rail	1.39	1.39	2.77
slide bar	0.04	0.04	0.07
Miscellaneous:			
switch set on/off	0.04	0.04	0.09
screw, loosen/tighten	0.12	0.08	0.21
tool, get/aside	0.04	0.04	0.08
walk, per pace, to/from	0.01	0.01	0.02

Appendix H - Worksheet for SCIACT replacement times

SCIACT Table

Unit	Rack	q - r conn	thrd conn	paces	hngd door	knrl scr	ret & nut	lsn scr	rem scr	knrl nuts	slide	Ckt Bds	Mod RR	QR fsnr	Cable rel	Sw settg	G, As tool	Nuts wsh	Rails	Total Min.	Total Sec.
400 hz PS	1&2		2	10	2	2			4		1		1	1			1		1	8.53	512
NB Modem	1&2	4	5		1	2	1						1							7.48	449
NB R/T	1&2	8	2		1	2				2			1		1					7.21	433
WB Modem	1&2	3	3		1	2	1						1							5.77	346
WB R/T	1&2	8	2		1	2				2			1		1					7.21	433
NBX,Y,Z Modem	4&5				1			6			1	1		10		1	2			3.48	209
NBX,Y,Z R/T	4&5					4					1	1		13	1	1	1			3.57	214
Comsec Ctl	6	2						4	4		1		1				2			4.90	294
DCSU WB1	6	5				4			4	2	1		1				1			6.40	384
Lambda PS	6&8	9		36					9		2		2				2		1	13.20	792
WB1,2 KG34	6					4					1		1							2.89	173
Floppy drive	7		2	16				6			1						1			3.44	207
NBX,Y,Z Modem Ctl	7					4					1	1		10			1			2.90	174
CP Synch'r	8	5	1			4		6			1		1	6			2			7.34	441
NBA,B Modem Ctl	8	1				4		4			1		1	4			2			4.67	280
R/T Sat Ctl (NBa,b;WB1,2)	8	4				4		4			1		1	4			2			5.67	340
EAM Alarm	-		1					4	4		1		1				2			4.92	295
KDU	-	2											2				1	1		5.23	314
Printer	-		2					4	1							1	2	4		7.35	441
REMOVE ELEMENT		RCQR	RCT	H	ODH	LSK	RET	LSP	RSP	LSK	ASL	RCB	RMOD	LFQR	RET	ACT	GAT	RNUT	RAIL		
REPLACE ELEMENT		ACQR	ACT	H	CDH	TSK	RET	TSP	ASP	TSK	ASL	ACB	AMOD	TFQR	RET	ACT	GAT	ANUT	RAIL		
REMOVE TIME (TMU)		76	411	18	148	202	186	206	239	202	59	317	1080	100	186	74	64	793	2309		
REPLACE TIME (TMU)		480	740	18	55	188	186	136	302	188	59	179	1496	97	186	74	64	975	2309		
TOTAL (TMU)		556	1151	36	203	390	372	342	541	390	118	496	2576	197	372	148	128	1768	4618		
TOTAL (Min.)		0.33	0.69	0.02	0.12	0.23	0.22	0.21	0.32	0.23	0.07	0.3	1.55	0.12	0.22	0.09	0.08	1.06	2.77		

Notes:
1. all Units have a constant time of 566 TMU for shut-down & start up & walking to bench
2. Lambda ps - 16 paces to walk around to back + equiv. of 5 x 4 paces for wt. to carry to table
- also allowed equiv. of 2 Module replacements on Lambda ps to allow for wt.

DISTRIBUTION

Dr. Earl Alluisi
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Perceptual Science Program
Office of Naval Research
Code 1142PS
800 North Quincy Street
Arlington, VA 22217-5000 (3 copies)

Computer Sciences Division
Code 1133
Office Of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Commanding Officer
Naval Medical R&D command
Bethesda, MD 20814-5055

Dr. Randall P. Schumaker
NRL A.I. Center
Code 7510
Naval Research Laboratory
Washington, D.C. 20375-5000

CDR. Thomas Jones
Code 125
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Special Assistant for Marine
Corps Matters
Code OOMC
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Director
Technical Information Division
Code 2627
Naval Research Laboratory
Washington, D.C. 20375-5000

Gerald S. Malecki
Office of Naval Research
Branch Office London
Box 39
FPO New York 09510-0700

Mr. John Davis
Combat Control System Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Systems Center
Orlando, FL 32813

Mr. Norm Beck
Combat Control Systems Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Dr. J.S Lawson, Jr.
4773-C Kahala Avenue
Honolulu, HI 96816

Mr. I. Lemaine
Engineering & Computer Science
Code 09
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Paul Heckman
Naval Ocean Systems Center
San Diego, CA 92152

Human Factors Engineering
Code 441
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dr. A.F. Norcio
Computer Sciences & Systems
Code 7592
Naval Research Laboratory
Washington, D.C. 20375-5000

Commander
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, D.C. 20361

LCDR T.Singer
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Dr. Jeff Grossman
Planning & Integration
Code 4403
Naval Ocean Systems Center
San Diego, CA 92152

LT. Dennis McBride
Human Factors Branch
Pacific Missle Test Center
Point Mugu., CA 93042

CDR.R. Carter
Office of Chief of Naval Operations
(OP-01B)
Washington, D.C. 20350

Dr. Kenneth R. Boff
AF AMRL/HE
Wright-Patterson AFB,OH 45433

Dr. L. Chmura
Compter Sciences & Systems
Code 7592
Naval Research Laboratory
Washington, D.C. 20375-5000

Dr. Michael Letsky
Office of the Chief of Naval Operations (OP-01B7)
Washington, D.C. 20350

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 North Quincy Street
Arlington, VA 22217-5000

Aircrew Systems Branch
Systems Engineering Test Directorate
US. Naval Test Center
Patuxent River, MD 20670

Mr. Milton Essoglou
Naval Facilities Engineering Command
R&D Plans and Programs
Code 03T
Hoffman Building II
Alexandria, VA 22332

Human Factors Engineering Branch
Naval Submarine Base
Submarine Medical Research Lab.
Groton, CT 06340

Commanding Officer
Naval Health Research Center
San Diego, CA 92152

D. Jerry Tobias
Auditory Research Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Mr. Mel Nunn
Test Technology Division, Code 9304
Naval Ocean Systems Center
San Diego, CA 92152

CAPT. W. Moroney
Naval Air Development Center
Code 602
Warminster, PA 18974

Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Dr. Robert A. Fleming
Human Factors Support Group
Naval Personnel Research & Development Ctr.
1411 South Fern Street
Arlington, VA 22202

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
Alexandria, VA 22333-5600

Technical Director
U.S Army Human Engineering Laboratory
Aberdeen Proving Ground MD 21105

Dr. A. Fregly
U.S. Air Force Office of Scientific Research
Life Science Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332-6448

Mr. Charles Bates, Director
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Dr. J. Tangney
Directorate Life Sciences
AFSOR
Bolling AFB
Washington, D.C. 20032-6448

Mr. Yale Smith
Rome Air Development
Center, RADC/COAD
Griffiss AFB
New York, NY 13441-5700

Director, Basic Research
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 *

*2 copies

Dr. A.D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF England

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Tech. Estab.
Teddington, Middlesex
TW11 OLN England

Dr. T.B. Sheridan
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Harold Van Cott
NAS-National Research Council
(COHF)
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. James H. Howard, Jr.
Department of Psychology
Catholic University
Washington, D.C. 20064

Dr. Christopher Wickens
Department of Psychology
University of Illinois
Urbana, IL 61801

Dr. Harry Snyder
Dept. of Industrial Engineering
Virginia Polytechnic Institute
and State University
Blacksburg, VA 24061

Dr. Robert Wherry
Analytics, Inc.
2500 Maryland Road
Willow Grove, PA 19090

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. William B. Rouse
School of Industrial and Systems
Georgia Institute of Technology
Atlanta, GA 30332

Dr. Hillel Einhorn
Graduate School of Business
University of Chicago
1101 E. 58th Street
Chicago, IL 60637

Dr. Douglas Towne
University of Southern California
Behavioral Technology laboratories
1845 S. Elena Ave., 4th Flr.
Redondo Beach, CA 90277

Dr. John Payne
Graduate School of Business Administration
Duke University
Durham, NC 27706

Paul Christopher
University of Southern California
University Library, mc-0183
University Park
Los Angles, CA 90089-0183

Dr. Paul Slovic
Decision Research
1201 Oak Street
Eugene, OR 97401

Dr. Nicholas A. Bond
4701 Anne Way
Carmichael, CA 95608

Dr. Jesse Orlansky
Institute of Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311